



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO.  | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|--|-------------|----------------------|---------------------|------------------|
| 10/766,545   | 01/27/2004  | Ylian Saint-Hilaire  | 1103-49-133008      | 3084             |
| 25943 7590 04/29/2008<br>SCHWABE, WILLIAMSON & WYATT, P.C.<br>PACWEST CENTER, SUITE 1900<br>1211 SW FIFTH AVENUE<br>PORTLAND, OR 97204 |             |                      |                     |                  |
| EXAMINER   |             |                      |                     |                  |
| ARCOS, CAROLINE H  |             |                      |                     |                  |
| ART UNIT   |             | PAPER NUMBER         |                     |                  |
| 2195   |             |                      |                     |                  |
| MAIL DATE  |             | DELIVERY MODE        |                     |                  |
| 04/29/2008   |             | PAPER                |                     |                  |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/766,545

**Applicant(s)**

SAINT-HILAIRE ET AL.

**Examiner**

Caroline Arcos

**Art Unit**

2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 27 January 2004.  
2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.  
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-31 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.  
6) ☒ Claim(s) 1-31 is/are rejected.  
7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.  
8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.  
10) ☒ The drawing(s) filed on 27 January 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)  
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3) ☒ Information Disclosure Statement(s) (PTO-8508)  
Paper No(s)/Mail Date 05/03/2004  
4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_  
5) ☐ Notice of Informal Patent Application  
6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. Claims 1-31 are pending for examination.

#### ***Claim Objections***

2. Claims 1-13 are objected to because of the following informalities: "itself" is a pronoun. Pronoun is not acceptable in a claimed language, only what is referred by "itself" should set forth in the claims appropriate correction is required.

#### ***Claim Rejections - 35 USC § 101***

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 25-31 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non statutory subject matter.
5. Claims 25 and 29 are rejected under 35 U.S.C. 101 because the claimed invention is directed to a system comprising at least one device; however, there is not enough support in the claim to suggest that the at least one device is of physical hardware nor execution of the applications on a hardware device. A virtual device or machine has the capability of hosting applications and where a virtual machine is software and software alone is directed to non- statutory subject matter. Claims 26-28 and 30-31 are rejected for similar reasons as discussed for their respective parent claims, as they fail to present any

limitation that resolve the deficiencies of the claims from which they depend.

*Claim Rejections - 35 USC § 112*

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 1-31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

- a. The following terms lacks antecedent basis:
  - i. The first set, the second set, the third set – claim 20.
  - ii. The first set, the second set, the third set – claim 29.
- b. The claim language in the following claims is not clearly understood:
  - iii. As per claim 1, line3, it is not clearly understood what is meant "logically chained to itself"? (i.e. looping). it is unclear what does "itself" refers to? (i.e. is it the second module or the thread?). Line 4, it is not clearly understood the execution of the second module will be as the same thread of which module or thread?(i.e. as the same thread of the first module). It is unclear whether the first module is in the same thread as the second module or two different threads? Line 5, it is unclear what are the criteria for causing the second module to be dynamically and logically chained? Line 6, it is not clearly understood the execution of the second

module will be as the same thread of which module or thread? (i.e. as the same thread of the first module). It is unclear how the registration of first event of interest is done? Line 8, it is unclear who is waiting for the notification of occurrence of event?(i.e. is it the first module or the second module or both?) Line 5, it is unclear what are the criteria for causing the second module to be dynamically and logically chained? And it is uncertain whether “causing” is the same as “causing” referred to in line 6 or different causing? It is unclear whether the second module gets chained dynamically and logically for the second time or only once. Line 10, it is not clearly understood the chaining of the second module will be as the same thread of which module or thread? (i.e. as the same thread of the first module).

iv. As per claim 2, line 3, it is unclear how the registration is performed? (i.e. record the events?). Line 5, it is unclear whether the first module cause the second module to be chained to itself for the third time or it is only one time the second module is being chained to itself? Line 6, it is unclear what does “itself” refers to? (i.e. is it the second module or the thread?).

v. As per claim 3, line 2, it is unclear what is causing and what are the criteria for the second module to be chained as part of the same thread. Line 3, it has the same deficiency as line 6 in claim 1. Line 4, it is uncertain whether “an event notification service” is the same as “an event notification service” in claim 1? (i.e. if it is the same event notification

service, it should be referred to as said event notification service). Line 5, line 7, it is unclear what is meant by “orchestrate”? Line 9, it is unclear from where the first pointer is retrieved?

vi. As per claim 4, line 3, it is unclear whether “an occurred one” is the same as “an occurred one” referred in claim 1? (i.e. if it is the same occurred one, it should be “referred to as said occurred one”).

vii. As per claim 5, it has the same deficiency as claim 4. line 4, line 7, it is unclear what is meant by “orchestrate”? Line 5, it is unclear whether “an occurred event” is the same as “an occurred one of the first events” referred in line 3 or is a different event?

viii. As per claim 6, line 2, it is unclear how the thread determination is detected? (i.e. insert a flush instruction).

ix. As per claim 7, line 7, it is unclear how the detection for thread termination is done. Line 11, line 7, it is unclear what is meant by “orchestrate”?

x. As per claim 8, lines 2-3, it is unclear what are the criteria causing the third module to be logically chained to the first and second modules? Line 4, it is unclear “same thread” as who? (i.e. first module, second module, different module?) Line 5, it is uncertain whether this is the second time the 3rd module is chained or only once. Line 8, it has the same deficiency as line 5. Line 11, it is unclear what is the relation between the “first events” and “second events”.

- xi. As per claim 9, it has the same deficiency as claim 1. Line 10, it is unclear how the detection of thread determination is done?
- xii. As per claim 10, line 3, it is unclear whether "an occurred one of the first events" is the same as "an occurred one of the first events" referred in claim 9. Line 10, it has the same deficiency as line 3. Line 12 it is unclear whether "an occurred event" is the same as "an occurred one of the first events" referred in line 3 or is a different event?
- xiii. As per claim 11, line 8, it is unclear how the detection for thread termination is done.
- xiv. As per claim 12, line 9, it is unclear how the detection for thread termination is done.
- xv. As per claim 13, lines 2-3, it is unclear what are the criteria causing the third module to be logically chained to the first and second modules? Line 4, it is unclear "same thread" as who? (i.e. first module, second module, different module?) Line 5, it is uncertain whether this is the second time the 3rd module is chained or only once. Line 8, it is unclear what is the relation between the "first events" and "second events"?
- xvi. As per claim 14, line 7, it is unclear what is meant by "orchestrated"?
- xvii. As per claim 15, line 2, it is unclear whether "a thread control data structure" is the same as "a thread control data structure" referred in claim 14? (i.e. if it is the same it should be referred to as said thread control data

structure). Lines 3-4, it is unclear whether “a single thread” is the same as “a single thread” referred in claim 14? (i.e. if it is the same it should be referred to as said single thread). Lines 5, . Lines 5, it is unclear whether “a thread” is the same as single thread or different thread?

xviii. As per claim 17, line 4, 7 and 9, it is unclear what is meant by “orchestrate”?

xix. As per claim 18, it is unclear what is meant by “UPNP”?

xx. As per claim 21, it has the same deficiency as claim 18.

xxi. As per claim 28, it has the same deficiency as claim 18.

xxii. As per claim 31, it has the same deficiency as claim 31.

xxiii. As per claim 19, line 6, it is unclear what are the criteria for selecting one of a set of executable instructions? Line 7, it is unclear whether “a set of executable instructions” is the same as “a set of executable instructions” referred in line 6? (i.e. if it is the same it should be referred to as said set of executable instructions).

xxiv. As per claim 20, it is unclear what is meant by “one of a first point”?

xxv. As per claim 22, line 8, it is unclear what is meant by “orchestrated”?

xxvi. As per claim 23, Line 2, it is unclear whether “a thread control data structure” is the same as “a thread control data structure” referred in claim 22? (i.e. if it is the same it should be referred to as said thread control data structure). Lines 3, it is unclear whether “a single thread” is the same as “a



single thread” referred in claim 22? (i.e. if it is the same it should be referred to as said single thread). Lines 5, it is unclear whether “a thread” is the same as single thread or different thread?

xxvii. As per claim 25, it has the same deficiency as claim 23.

xxviii. As per claim 24, line 3, 6 and 8, it is unclear what is meant by “orchestrate”?

xxix. As per claim 27, it has the same deficiency as claim 24.

xxx. As per claim 28, it has the same deficiency as claim 19. Lines 5, it is unclear whether “a single thread” is the same as “a single thread” referred in claim 25? (i.e. if it is the same it should be referred to as said single thread).

### *Claim Rejections - 35 USC § 102*

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-17, 19-20, and 22-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Wang et al. (US 2002/0144083 A1).

10. As per claim 1, Wang teaches the invention as claimed including a method comprising:

a first module of a thread in a computing environment dynamically causing a second module to be logically chained to itself, enabling the second module to be executed as part of the same thread (par. [0047], lines 1-20; par. [0048], lines 1-4; par. [0061]; par. [0074]);

causing the second module dynamically and logically chained as part of the same thread to register first events of interest to the second module with an event notification service of the computing environment (par.[0047], lines 6-20);

waiting for notification of occurrence of one or more of the first events (par. [0055]; par. [0085], lines 12-20); and

causing the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events (par. [0074]).

11. As per claim 2, Wang teaches that the second module comprises a first set of executable instructions designed to register the first events with said event notification service, and a first pointer to the first set of executable instructions (par. [0048], lines 1-5; par. [0049], lines 1-2; par. [0085], lines 5-8);

the first module dynamically causing the second module to be logically chained to itself, by invoking a thread module chaining service of the computing environment (par.[0047], line 15-20);

The method further comprises the thread module chaining service annotating a control data structure of the thread to logically associate the second module with the first module, including with the annotation, the first pointer to the first set of executable instructions of the second module (par.[0047], line 15-20; par. [0048], lines 1-5; par.

[0063], lines 1-5).

12. As per claim 3, Wang teaches that said causing of the second module dynamically and logically chained as part of the same thread to register first events of interest to the second module with an event notification service of the computing environment comprises the first module invoking the thread module chaining service to orchestrate registration of events of interest to logically chained modules of the thread with the event notification service (par. [0047], lines 1-8; par. [0047], lines 15-20); and

the method further comprises the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions ([par. 0048]; par. [0049]).

13. As per claim 4, Wang teaches the second module further comprises a second set of executable instructions designed to process an occurred one of the first events, and a second pointer to the second set of executable instructions (par. [0085], lines 5-20); and

the thread module chaining service further includes with the annotation, the second pointer to the second set of executable instructions of the second module (par. [0085]).

14. As per claim 5, Wang teaches said causing of the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events comprises the first module invoking the thread module chaining service to orchestrate

processing of an occurred event by the logically chained modules of the thread (par.

[0047], lines 1-20; par. [0048], lines 1-4; par. [0061]; par. [0074]); and

the method further comprises the thread module chaining service retrieving the second pointer to the second set of executable instructions, and causing the second set of executable instructions to be executed, using the second pointer to locate the second set of executable instructions(par. [0085]).

15. As per claim 6; Wang teaches that detecting for a thread termination condition ([par. [0081], lines 18-20); and causing the thread to be terminated after detecting the thread termination condition (par. [0081], lines 20-22).

16. As per claim 7, Wang teaches that the second module further comprises a third set of executable instructions designed to perform termination clean up for the second module, and a third pointer to the third set of executable instructions; the thread module chaining service further includes with the annotation, the third pointer to the third set of executable instructions of the second module (par. [0081], lines 16-22; par. [0083]; par. [0085], lines 1-20).

said detecting comprises the first module detecting for the thread termination condition ([par. [0081], lines 18-20);

said causing of the thread to be terminated after detecting the thread termination condition comprises the first module invoking the thread module chaining service to orchestrate thread termination clean up by the logically chained modules of the thread(par. [0081], lines 20-22); and

the method further comprises the thread module chaining service retrieving the third pointer to the third set of executable instructions, and causing the third set of executable instructions to be executed, using the third pointer to locate the third set of executable instructions (par. [0085], lines 1-9).

17. As per claim 8, Wang teaches that the first module dynamically causing a third module to be logically chained to the first and second modules, enabling the third module to be executed as part of the same thread (par. [0047], lines 18-20);

causing the third module dynamically and logically chained as part of the same thread to register second events of interest to the third module with the event notification service of the computing environment (par. [0047]; par. [0075]); and

causing the dynamically and logically chained third module as part of the same thread to process an occurred one of the second events, said waiting for notification of occurrence further comprising waiting for notification of one or more of the second events (par. [0047], lines 4-14; par. [0055]; claim 15).

18. As per claim 9, Wang teaches a method comprising:

a first module of a thread in a computing environment dynamically causing a second module to be logically chained to itself, enabling the second module to be executed as part of the same thread (par. [0048], lines 1-5; par. [0049], lines 1-2);

waiting for notification of occurrence of one or more of first events of interest to the second module dynamically and logically chained as part of the same thread(par. [0055]; par. [0074]; par. [0085], lines 12-20);

causing the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events (par. [0047], lines 6-20);

detecting for a thread termination condition ([par. [0081], lines 18-20); and  
causing the thread to be terminated after detecting the thread termination condition (par. [0081], lines 20-22).

19. As per claim 10, Wang teaches that second module comprises a first set of executable instructions designed to process an occurred one of the first events, and a first pointer to the first set of executable instructions (par. [0048], lines 1-5; par. [0049], lines 1-2);

the method further comprises a thread module chaining service annotating a control data structure of the thread to logically associate the second module with the first module, including with the annotation, the first pointer to the first set of executable instructions of the second module (par.[0047], line 15-20; par. [0048], lines 1-5; par. [0063], lines 1-5);

said causing of the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events comprises the first module invoking the thread module chaining service to orchestrate processing of an occurred event by logically chained modules of the thread(par.[0047], lines 6-20); and

the method further comprises the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of

executable instructions([par. 0048]; par. [0049]).

20. As per claim 11, Wang teaches that the second module further comprises a second set of executable instructions designed to perform termination clean up for the second module, and a second pointer to the second set of executable instructions (par. [0081], lines 18-20; par. [0085], Lines 1-20);

the thread module chaining service further includes with the annotation, the second pointer to the second set of executable instructions of the second module (par. [0085], lines 1-9);

said detecting comprises the first module detecting for the thread termination condition ([par. [0081], lines 18-20);

said causing of the thread to be terminated after detecting the thread termination condition comprises the first module invoking the thread module chaining service to orchestrate thread termination clean up by the logically chained modules of the thread(par. [0081], lines 20-22; par. [0083]);  
and

the method further comprises the thread module chaining service retrieving the second pointer to the second set of executable instructions, and causing the second set of executable instructions to be executed, using the second pointer to locate the second set of executable instructions (par. [0083]; par. [0085]).

21. As per claim 12, Wang teaches that the second module comprises a first set of executable instructions designed to perform termination clean up for the second module,

and a first pointer to the first set of executable instructions ([0085]; par.[0081], lines 16-22; par.[0083]);

the method further comprises a thread module chaining service annotating a control data structure of the thread to logically associate the second module with the first module, including with the annotation, the first pointer to the first set of executable instructions of the second module (par.[0084], lines 1-5; par. [0049], lines 1-2; par. [0085], lines 1-9);

said detecting comprises the first module detecting for the thread termination condition ([par. [0081], lines 18-20; par. [0083]);

said causing of the thread to be terminated after detecting the thread termination condition comprises the first module invoking the thread module chaining service to orchestrate thread termination clean up by logically chained modules of the thread (par. [0081], lines 18-20; par. [0083]); and

the method further comprises the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the second set of executable instructions (par. [0085]).

22. As per claim 13, Wang teaches that the first module dynamically causing a third module to be logically chained to the first and second modules, enabling the third module to be executed as part of the same thread (par. [0047], lines 15-21; par. [0061] and

causing the third module dynamically and logically chained as part of the same thread to process an occurred one of second events (par. [0047], lines 1-11), said waiting



for notification of occurrence of one or more of first events further comprising waiting  
for notification of one or more of the second events (par.0085], lines 9-20).

23. As per claim 14, Wang teaches a computing device comprising:

storage medium having stored therein a first plurality of executable instructions designed to provide a thread module chaining service to facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread on the computing device, including an ability to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread (par. [0086], lines 6-9; par. [0047], lines 1-11; par. [0048], lines 1-5); and

at least one processor coupled to the storage medium to execute the instructions (par. [0083], lines 1-2).

24. As per claim 15, Wang teaches the ability to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread includes an ability to annotate the thread control data structure with control data about a module to be logically chained to be a part of a thread, when dynamically invoked to logically chain the module to be a part of the thread (par. [0086], lines 6-9; par. [0047], lines 1-11; par. [0048]).

25. As per claim 16, Wang teaches the control data includes at least a selected one of a pointer of the module pointing to a plurality of executable instructions of the

module to register events of interest to the module with an event notification service of the computing device (par. [0047], lines 1-20; par. [0085], lines 1-9),

    a pointer of the module pointing to a plurality of executable instructions of the module to process an occurred event of interest to the module,(par. [0047], line1-8),

    a pointer of the module pointing to a plurality of executable instructions of the module to perform thread termination clean up for the module, and the ability includes an ability to extract the selected one or more pointers from the module(par. [0081], lines 18-20; par. [0083]; par. [0085]).

26. As per claim 17, Wang teaches that the first plurality of executable instructions further provide the thread module chaining service with at least a selected one of

    an ability to orchestrate registration of events of interest to the logically chained modules with an event notification service by the logically chained modules (par. [0047], lines 1-20; par. [0055], lines 1-13),

    an ability to orchestrate processing of an occurred event of interest by one or more of the logically chained modules (par. [0047], lines 1-20; par. [0085], lines 1-9), and

    an ability to orchestrate thread termination clean up of the logically chained modules by the respective logically chained modules ([par. [0081], lines 18-20; par. [0083]).

27. As per claim 19, Wang teaches a computing device comprising: storage medium having stored therein a first and a second module, with the first module equipped to logically chain the second module to the first module dynamically during execution of

the first module, enabling the second module to execute with the first module as a single thread, and the second module having at least a selected one of a set of executable instructions to register events of interest to the second module, a set of executable instructions to process an occurred one of the events of interest, and a set of executable instructions to perform clean up during thread termination (par. [0086], lines 6-9; par. [0047], lines 1-11; par. [0048], lines 1-5; par. [0081], lines 18-20; par. [0083]); and  
at least one processor coupled to the storage medium to execute the instructions (par. [0083], lines 1-2).

28. As per claim 20, Wang teaches that the second module further includes at least a corresponding one of a first point, a second pointer, and a third pointer pointing to the first set, the second set, and the third set of executable instructions respectively (par. [0085]).

29. As per claim 22, Wang teaches an article of manufacture comprising: a computer readable medium (par.[0086], lines 5-9) ; and  
a plurality of executable instructions designed to implement a thread module chaining service to facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread in a computing environment, including an ability to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread(par. [0047]; par. [0048]).

30. As per claim 23, Wang teaches the ability to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread includes an ability to annotate the thread control data structure with control data about a module to be logically chained to be a part of a thread, when dynamically invoked to logically chain the module to be a part of the thread (par. [0047], lines 1-11; par. [0048]).

31. As per claim 24, Wang teaches the first plurality of executable instructions further provide the thread module chaining service with at least a selected one of

an ability to orchestrate registration of events of interest to the logically chained modules with an event notification service by the logically chained modules (par. [0047], lines 1-20; par. [0055], lines 1-13),

an ability to orchestrate processing of an occurred event of interest by one or more of the logically chained modules (par. [0047], lines 1-20; par. [0085], lines 1-9), and

an ability to orchestrate thread termination clean up of the logically chained modules by the respective logically chained modules ([par. [0081], lines 18-20; par. [0083]).

### ***Claim Rejections - 35 USC § 103***

32. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

33. Claims 25-27, 29-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US 2002/0144083 A1).

34. As per claim 25, Wang teaches a system comprising:

a first device including a plurality of executable instructions designed to implement a thread module chaining service to facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread in the first device, including an ability to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread(par. [0047]; par. [0048]; par. [0086]).

35. Wang doesn't explicitly teach that a second device coupled to the first device. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to couple a second device to the first device would improve system efficiency by having a direct communication/ link between two device allows faster data access thus improve system performance.

36. As per claim 26 Wang teaches the ability of the thread module chaining service to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread includes an ability to annotate the thread control data structure with control data

about a module to be logically chained to be a part of a thread, when dynamically invoked to logically chain the module to be a part of the thread (par. [0047]; par. [0048]).

37. As per claim 27, Wang teaches the plurality of executable instructions further provide the thread module chaining service with at least a selected one of

an ability to orchestrate registration of events of interest to the logically chained modules with an event notification service by the logically chained modules(par. [0047], lines 1-20; par. [0055], lines 1-13),

an ability to orchestrate processing of an occurred event of interest by one or more of the logically chained modules (par. [0047], lines 1-20; par. [0055], lines 1-13), and

an ability to orchestrate thread termination clean up of the logically chained modules by the respective logically chained modules ([par. [0081], lines 18-20; par. [0083]).

38. As per claim 29, Wang teaches a system comprising:

a first device having a first and a second module, with the first module equipped to logically chain the second module to the first module dynamically during execution of the first module, enabling the second module to execute with the first module as a single thread, and the second module having at least a selected one of a set of executable instructions to register events of interest to the second module, a set of executable instructions to process an occurred one of the events of interest, and a set of executable instructions to perform clean up during thread termination(par. [0047]; par. [0048] [par.

[0081], lines 18-20; par. [0083]; par. [0086]).

39. Wang doesn't explicitly teach that a second device coupled to the first device. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to couple a second device to the first device would improve system efficiency by having a direct communication/ link between two device allows faster data access thus improve system performance.

40. As per claim 30, Wang teaches wherein the second module further includes at least a corresponding one of a first point, a second pointer, and a third pointer pointing to the first set, the second set, and the third set of executable instructions respectively(par. [0085]).

41. Claims 18, 21, 28 and 31 are rejected under 35 U.S.C. 102(b) as being anticipated by Wang et al. (US 2002/0144083 A1), in view of UPNP ("UPNP AV architecture: 0.83", UPNP forum, Version 1.0, 2002, pages 1-22).

42. As per claim 18, Wang doesn't teach that the computing device comprises a UPNP control point. However, UPNP teaches that the computing device comprises a UPNP control point (fig. 1; Fig. 2; pg. 4, lines 5-9).

43. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a UPNP control point would improve and simplify the

implementation of networks in the home (data sharing, communications, and entertainment) and corporate environments. It allows peripheral devices to discover and connect to other devices and to enumerate the characteristics of those devices.

44. As per claim 21, Wang doesn't teach that the computing device comprises a UPNP control point. However, UPNP teaches that the computing device comprises a UPNP control point (fig. 1; Fig. 2; pg. 4, lines 5-9).

45. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a UPNP control point would improve and simplify the implementation of networks in the home (data sharing, communications, and entertainment) and corporate environments. It allows peripheral devices to discover and connect to other devices and to enumerate the characteristics of those devices.

46. As per claim 28, Wang doesn't teach that the first computing device comprises a UPNP control point, and the second computing device comprises a selected one of a UPNP server and a UPNP renderer. However, UPNP teaches that the first computing device comprises a UPNP control point, and the second computing device comprises a selected one of a UPNP server and a UPNP renderer (pg. 5, lines 1-17).

47. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include UPNP control point in the first device and one of a UPNP server and a UPNP renderer in the second device would improve the system performance



by facilitating rendering individual items of content on a specific rendering device.

48. As per claim 31. Wang doesn't teach that the first computing device comprises a UPNP control point, and the second computing device comprises a selected one of a UPNP server and a UPNP renderer. However, UPNP teaches that the first computing device comprises a UPNP control point, and the second computing device comprises a selected one of a UPNP server and a UPNP renderer (pg. 5, lines 1-17).

49. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include UPNP control point in the first device and one of a UPNP server and a UPNP renderer in the second device would improve the system performance by facilitating rendering individual items of content on a specific rendering device.

### ***Conclusion***

50. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

(US 5925108 A) teaches Event notification in a computer system.

(US 7111305 B2) teaches Facilitating event notification through use of an inverse mapping structure for subset determination.

(US 5828882 A) teaches Event notification facility.

51. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Caroline Arcos whose telephone number is 571-270-3151.

The examiner can normally be reached on Monday-Thursday 7:00 AM to 5:30 PM.

52. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

53. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Patent examiner  
Caroline Arcos

/Meng-Ai An/  
Supervisory Patent Examiner, Art Unit 2195